



# Exploiting Extensive External Information for Event Detection Through Semantic Networks Word Representation and Attention Map

Zechen Wang<sup>1</sup>, Shupeng Wang<sup>2(✉)</sup>, Lei Zhang<sup>2(✉)</sup>, and Yong Wang<sup>2</sup>

<sup>1</sup> School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China  
wangzechen@iie.ac.cn

<sup>2</sup> Institute of Information Engineering, CAS, Beijing, China  
{wangshupeng, zhanglei1, wangyong}@iie.ac.cn

**Abstract.** Event detection is one of the key tasks to construct knowledge graph and reason graph, also a hot and difficult problem in information extraction. Automatic event detection from unstructured natural language text has far-reaching significance for human cognition and intelligent analysis. However, limited by the source and genre, corpora for event detection can not provide enough information to solve the problems of polysemy, synonym association and lack of information. To solve these problems, this paper proposes a brand new Event Detection model based on Extensive External Information (EDEEI). The model employs external corpus, semantic network, part of speech and attention map to extract complete and accurate triggers. Experiments on ACE 2005 benchmark dataset show that the model effectively uses the external knowledge to detect events, and is significantly superior to the state-of-the-art event detection methods.

**Keywords:** Event detection · External information · Wnet2vec · Semantic network · Attention mechanism

## 1 Introduction

Event detection mines the necessary information from unstructured data to represent events structurally with 5W1H framework [21]. An event is composed of a trigger which indicates the existence of the event, and several arguments which constitute the detail information [9]. Thus, the target of event detection is to detect whether trigger existing in the sentence, and determine which event type it belongs to.

Event detection is conducive to the storage, retrieval, representation and analysis of event information. Although event detection has been widely used in various areas, [17] such as abnormal public opinion detection [11, 20], news recommendation systems [7, 19], risk analysis application [1], monitoring system [14] and decision support tools [4], it still faces tremendous difficulties and challenges. There are great obstacles in event detection. Especially, it is difficult to deal with words with multiple meanings on the basis of the information contained in the local context. For example, in the field of financial analysis, an investment event can be detected from the sentence “The charity

was prospected of giving money from a bank.” It can attribute to three problems in event detection tasks as follows: First, **Polysemy**. In different languages, a word may represent different types of entities in different contexts. Thus, the meaning of “bank” can not be judged directly in the sentence from the above example. Consequently, the wrong judgment causes the problem that polysemous word can not be extracted as the correct trigger. Finally, the event element related to investment would be wrongly judged, which may result in the loss of important elements in further analysis and eventually cause unwise investment decision. Second, **Synonym Association**. Some word may only appear once in a corpus, but it may have several synonyms in the same corpus. Hence, it is difficult to establish a relationship between synonyms only through local corpus. For example, the word “giving” in the above example is synonymous with the verb “investing” which expresses investment behavior. Yet these words do not appear in the same context, so the words like “investing” can not be directly associated with the word “giving”. The word “giving” dose not always represent a investment-related behavior, but may have other meanings such as “present” and “supply”. Therefore, if it does not associate with words like “investing”, it is nearly impossible to classify the word as a trigger of investment event. Third, **Lack of Information**. The context usually includes word information, word position information and so on. More seriously, the information does not contain part of speech information. Nonetheless, it is crucial to identify and classify triggers in the event detection task. But if part of speech information is not used sufficiently, it would be difficult to extract triggers which tend to hide in nouns, verbs and adjectives. It will reduce the accuracy of the extraction of triggers.

Therefore, the above three problems need to be solved at the same time to detect events completely and accurately.

## 2 Related Works

In recent years, with the increasingly wide application of event detection, there are some related research works. The existing studies are in mainly three categories according to the problems they solve: polysemy [3,6], synonym association [8,12] and lack of information [16].

The problem of polysemy is mainly reflected in the stage of word expression. Various word embedding models, such as CBOW and skip-gram, have been proposed. These models can not generate different vectors for different meanings of the same word. At present, the methods to solve the problem of polysemy include: 1) The methods cluster the meanings of the same word corresponding to different contexts to distinguish the meanings of the word [3]; 2) The methods use cross language information. These methods translate different meanings of the same word into other languages to make each meaning of the word corresponding to one word in the target language [6]. But these methods have a disadvantage that a word is fixed to one word vector, no matter how many meanings the word has. Accordingly, the meanings of a word can not be distinguished. As a result, this problem can not be well solved.

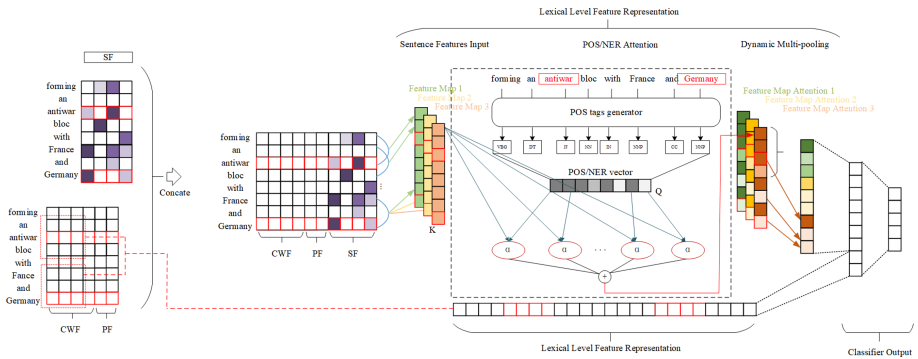
The methods aim at solving the problem of synonym association mainly describe the association between synonyms by using rules directly according to dictionaries or through external corpus information. Synonym association plays an important role in

event detection task, especially in the event type classification mentioned above. Synonyms can be associated on the basis of event-trigger dictionaries [8] or synonym sets according to semantic networks [12]. However, these methods need to construct a word list in advance to support event detection, and update the word list if corpus changes. Consequently, these methods can only be used in limited situations, and can not solve the problem of synonym association in broad areas.

To solve the problem of lack of information, many features such as vocabulary, syntax and semantics can be used as input. For example, an idea proposed by Liu et al. [16] claims that triggers and arguments should receive more attention than other words, so they construct gold attention vectors which only encode each trigger, argument and context word. Nevertheless, the constitution of gold attention vectors rely heavily on syntax knowledge and domain knowledge which can be used in event detection.

In summary, due to the limitation of the scope of prior knowledge, the studies on event detection till now can partly solve problems in event detection. Furthermore, they also can not solve the problems of polysemy, synonym association and lack of information at once within a single model.

### 3 Method



**Fig. 1.** The EDEEI model includes word vector representation, feature extraction, feature optimization, feature selection and classifier.

In this section, we present our framework for the proposed EDEEI algorithm. The proposed framework is illustrated in Fig. 1.

The word vector representation module generates three kinds of vectors: Content Word Features (CWF), Semantic Features (SF) and Position Features (PF). The newly-generated word vectors are used for feature extraction, feature optimization and feature selection. Feature optimization refined raw features from feature extraction, and provide processed features to feature selection to capture high-quality features. Finally, the features are inputted into the classifier to obtain the triggers and their classification.

**Word Vector Representation.** We derive a new word vector structure based on BERT [5] and wnet2vec [18]. To solve the problem of polysemy, the proposed framework utilizes BERT to generate the CWF, which identifies different word meanings of one word according to a variable external corpus. Wnet2vec is exploited in our model to generate SFs which can better express the semantic relationship between synonyms. Wnet2vec has the ability of generating word vectors through the transformation from semantic network to semantic space. Finally, the PF is defined, that is, the relative distance between the current word and the candidate trigger. In order to encode the PF, each distance value is represented by an embedded vector. Let the size of CWF be  $d_{CWF}$ , the size of SF be  $d_{SF}$ , and the size of position code be  $d_{PF}$ . Represent word vector of the  $i$ -th word in the sentence as  $x_i \in R^d$ ,  $d = d_{CWF} + d_{SF} + d_{PF} * 2$ . Represent sentence of length  $n$  as:  $x_{1:n} = x_1 \oplus x_2 \oplus \dots \oplus x_n$ , where  $\oplus$  is the concatenation operator. The vectors are spliced to form a matrix  $X \in R^{n \times d}$ .

**Feature Extraction.** The convolution layer is to capture and compress the semantics of the whole sentence, so as to extract these valuable semantics into feature maps. Each convolution operation contains a convolution kernel  $w \in R^{h \times d}$ , and the window size is  $h$ . For example, from the window size of  $x_{i:i+h-1}$ , the module generates the feature  $c_i$ :  $c_i = f(w \cdot x_{i:i+h-1} + b)$ , where  $b$  is the bias and  $f$  is the nonlinear activation function. Apply the calculation process to the sentence  $x_{i:n}$  to generate the feature map  $C = \{c_1, c_2, \dots, c_n\}$ . Corresponding to  $m$  convolution kernels  $W = w_1, w_2, \dots, w_m$ , the result of feature maps generated by  $M$  is expressed as  $\{C_1, C_2, \dots, C_m\}$ .

**Feature Optimization.** This module uses the POS tag generation tool provided by Stanford CoreNLP to annotate the sentences. One-hot coding is applied to POS tags of different types, and the coding vector is  $k_1$ . According to the coding of each POS tag, POS matrix is generated for sentences of length  $n$ , and the matrix size is  $M_{POS} \in R^{k_1 \times n}$ .

The crucial parts corresponding to specific POS tags are emphasised by attention mechanism. The feature optimization module takes POS tags and feature maps as the inputs of attention mechanism. Each feature map generated in feature extraction module is a vector of length  $n - h + 1$ . The feature maps represented as  $K$  are taken as the inputs of attention mechanism. The attention calculation process is as follows. A new random matrix  $WQ$  of length  $w$  is computed. The product of two vectors is calculated to obtain a new matrix  $Q$ . The random matrix  $WK$  whose width is  $w$  and the length is  $k_1$  is acquired, and the product of the random matrix  $WK$  and  $WQ$  produces the matrix  $WV$ . Calculate the product of  $WV$  and feature map to gain the matrix  $V$ .

Based on the three generated  $K, Q, V$  matrices, an attention matrix  $Z$  is calculated by using the following formula:  $Z = \text{softmax} \left( \frac{Q \times K^T}{\sqrt{n-h+1}} \right) V$ . Train  $Wk, WQ, WV$  matrices. The scoring function is as follows:  $f_{\text{score}} = \frac{Q \cdot K^T}{\sqrt{n-h+1}}$ . Then the  $Z$  matrix is compressed with max pooling to generate a vector  $z$ . Based on the updated  $WK, WQ$  and  $WV$ , the product of  $z$  and  $K$  constructs a new attention map with the size of  $n - h + 1$ .

**Feature Selection.** The feature selection module employs dynamic multi-pooling to further extract the valuable features. Furthermore, the features are concatenated to produce lexical level vectors which contain information of what role the current word plays in an event. The calculation process of dynamic multi-pooling is given as follows:  $[y_{1,p_t}]_i = \max \{ [C_1]_i, \dots, [C_{p_t}]_i \}$ ,  $[y_{p_t+1,p_n}]_i = \max \{ [C_{p_t+1}]_i, \dots, [C_n]_i \}$ , where  $[y]_i$  is the  $i$ -th value of the vector,  $p_t$  is the position of trigger  $t$ , and  $C_i$  is the  $i$ -th value in the attention map  $C$ . We use the maximum pooling result of each segment as the feature vector  $L$  at the sentence level.

**Classifier.** This module concatenates the CWFs of the current word and the words on the left and right of the current one, to obtain the vector  $P$  of length  $3 * CWF$ . The learned sentence level features and word features are connected into a vector  $F = [L, P]$ . In order to calculate the confidence of the event type of each trigger, the feature vector is inputted into the classifier  $O = W_s F + b_s$ .  $W_s$  is the transformation matrix of the classifier,  $b_s$  is the bias, and  $O$  is the final output of the network, where the output type is equal to the total number of event types plus one to include the “not a trigger” tag.

## 4 Experiment

The ACE 2005 is utilized as the benchmark experimental dataset. The test set used in the experiment contains 40 Newswire articles and 30 other documents randomly selected from different genres. The remaining 529 documents are used as the training set.

**Evaluation of Event Detection Methods.** To demonstrate how the proposed algorithm improves the performance over the state-of-the-art event detection methods, we compare the following representative methods from the literature:

- (1) Li’s baseline [13]: Li et al. proposed a feature-based system which used artificially designed lexical features, basic features and syntactic features.
- (2) Liao’s cross-event [15]: The cross-event detection method proposed by Liao and Grishman used document level information to improve the performance.
- (3) Hong’s cross-entity [10]: Hong et al. exploited a method to extract events through cross-entity reasoning.
- (4) Li’s joint model [13]: Li et al. also developed an event extraction method based on event structure prediction.
- (5) DMCNN method [2]: A framework based on dynamic multi-pooling convolutional neural network.

In all the methods, EDEEI model has achieved the best performance. Compared with the state-of-the-art methods, F value of trigger identification is significantly improved. The results displayed in Table 1 illustrate three important facts on the method. Firstly, it is necessary to solve the problems of polysemy, synonym association and lack of information at the same time. Secondly, the variable external knowledge can effectively improve the accuracy of event detection. Thirdly, the hierarchical detecting method with feature optimization can make event detection more completely and precisely.

**Table 1.** Overall performance on the ACE 2005 blind test data

Methods	Trigger identification			Trigger classification		
	P	R	F	P	R	F
Li's baseline	76.2	60.5	67.4	74.5	59.1	65.9
Liao's cross-event	N/A			68.7	68.9	68.8
Hong's cross-entity	N/A			72.9	64.3	68.3
Li's joint model	76.9	65	70.4	73.7	62.3	67.5
DMCNN model	80.4	67.7	73.5	75.6	63.6	69.1
EDEEI model	<b>77.0</b>	<b>72.9</b>	<b>74.9</b>	<b>77.3</b>	<b>63.2</b>	<b>69.5</b>

**Analysis of Different Word Vectors.** This section presents a detailed comparison of the word vectors generated from Word2vec, BERT, wnet2vec and BERT+wnet2vec respectively. The purpose is to test for advantages and disadvantages of BERT+wnet2vec approaches versus other word vectors under the task of event detection.

**Table 2.** Performance with different word vectors

Methods	Trigger identification F (%)	Trigger identification + classification F (%)
Word2Vec	72.2	65.9
BERT	72.7	66.7
wnet2vec	67.9	58.8
BERT+wnet2vec	<b>74.9</b>	<b>69.5</b>

The advantages of using BERT+wnet2vec can be observed visually and quantitatively in Table 2. It can be seen that the combination of BERT and wnet2vec achieves the best performance on both trigger identification and trigger classification.

In conclusion, traditional methods such as word2vec rely on a small corpus to generate word vectors, and can not solve the problem of polysemy and synonym association. Compared with word2vec, the combination of the two methods can obtain the best experimental effect. This proves that BERT+wnet2vec can effectively solve the problem of polysemy and synonym association.

## 5 Conclusion

This paper addresses three important problems in event detection: polysymy, synonym association and lack of information. In order to solve these problems, we propose a

brand new Event Detection model based on Extensive External Information (EDEEI), and give a novel method which involves external corpus, semantic network, part of speech and attention map to detect events completely and accurately. This framework can solve the above three problems at the same time. An attention mechanism with part of speech information is designed to optimize the extracted features and make the features related to triggers easier to capture. The experiments on widely used ACE 2005 benchmark dataset confirm that the proposed method significantly outperforms the existing state-of-the-art methods for event detection. Furthermore, we present numerous qualitative and quantitative analyses about experimental results. In the light of excellent performance and analyses, it is believed that the proposed algorithm can be a useful tool for event detection.

**Acknowledgments.** This work was supported by National Natural Science Foundation of China (No. 61931019) and National Key Research and Development Program Project (No. 2019QY2404).

## References

1. Chau, M.T., Esteves, D., Lehmann, J.: A neural-based model to predict the future natural gas market price through open-domain event extraction. *ESWC* **2611**, 17–31 (2020)
2. Chen, Y., Xu, L., Liu, K., Zeng, D., Zhao, J.: Event extraction via dynamic multi-pooling convolutional neural networks. In: *ACL*, pp. 167–176 (2015)
3. Chen, Y., Yang, H., Liu, K.: Collective event detection via a hierarchical and bias tagging networks with gated multi-level attention mechanisms. In: *EMNLP*, pp. 1267–1276 (2018)
4. Cheng, D., Yang, F., Wang, X.: Knowledge graph-based event embedding framework for financial quantitative investments. In: *ACM SIGIR*, pp. 2221–2230 (2020)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: *ACL*, pp. 4171–4186 (2019)
6. Ferguson, J., Lockard, C., Weld, D.S., Hajishirzi, H.: Semi-supervised event extraction with paraphrase clusters. In: *NAACL-HLT*, pp. 359–364 (2018)
7. George, S.K., Jagathy Raj, V.P., Gopalan, S.K.: Personalized news media extraction and archival framework with news ordering and localization. In: Tuba, M., Akashe, S., Joshi, A. (eds.) *Information and Communication Technology for Sustainable Development. AISC*, vol. 933, pp. 463–471. Springer, Singapore (2020). [https://doi.org/10.1007/978-981-13-7166-0\\_46](https://doi.org/10.1007/978-981-13-7166-0_46)
8. Han, S., Hao, X., Huang, H.: An event-extraction approach for business analysis from online Chinese news. *Electron. Commer. R A* **28**, 244–260 (2018)
9. Hogenboom, F., Frasinca, F., Kaymak, U., De Jong, F.: An overview of event extraction from text. In: *DeRiVE 2011*, vol. 779, pp. 48–57 (2011)
10. Hong, Y., Zhang, J., Ma, B., Yao, J., Zhou, G., Zhu, Q.: Using cross-entity inference to improve event extraction. In: *ACL*, pp. 1127–1136 (2011)
11. Huang, L., et al.: Liberal event extraction and event schema induction. In: *ACL*, pp. 258–268 (2016)
12. Iqbal, K., Khan, M.Y., Wasi, S., Mahboob, S., Ahmed, T.: On extraction of event information from social text streams: an unpretentious NLP solution. *IJCSNS* **19**(9), 1 (2019)
13. Li, Q., Ji, H., Huang, L.: Joint event extraction via structured prediction with global features. In: *ACL*, pp. 73–82 (2013)

14. Liang, Z., Pan, D., Deng, Y.: Research on the knowledge association reasoning of financial reports based on a graph network. *Sustainability* **12**(7), 2795 (2020)
15. Liao, S., Grishman, R.: Using document level cross-event inference to improve event extraction. In: *ACL*, pp. 789–797 (2010)
16. Liu, S., Chen, Y., Liu, K., Zhao, J.: Exploiting argument information to improve event detection via supervised attention mechanisms. In: *ACL*, pp. 1789–1798 (2017)
17. Mukhina, K., Visheratin, A., Nasonov, D.: Spatiotemporal filtering pipeline for efficient social networks data processing algorithms. In: *ICCS*, pp. 86–99 (2020)
18. Saedi, C., Branco, A., Rodrigues, J., Silva, J.: Wordnet embeddings. In: *RepL4NLP*, pp. 122–131 (2018)
19. Sheu, H.S., Li, S.: Context-aware graph embedding for session-based news recommendation. In: *RecSys*, pp. 657–662 (2020)
20. Wang, Z., Sun, L., Li, X., Wang, L.: Event extraction via dmcnn in open domain public sentiment information. In: *ICPCSEE*, pp. 90–100 (2020)
21. Xiang, W., Wang, B.: A survey of event extraction from text. *IEEE Access* **7**, 173111–173137 (2019)